Completing Wikipedia's Hyperlink Structure through Dimensionality Reduction

Robert West Doina Precup Joelle Pineau

School of Computer Science
McGill University
Montréal, Québec, Canada
{rwest, dprecup, jpineau}@cs.mcgill.ca

ABSTRACT

Wikipedia is the largest monolithic repository of human knowledge. In addition to its sheer size, it represents a new encyclopedic paradigm by interconnecting articles through hyperlinks. However, since these links are created by human authors, links one would expect to see are often missing. The goal of this work is to detect such gaps automatically. In this paper, we propose a novel method for augmenting the structure of hyperlinked document collections such as Wikipedia. It does not require the extraction of any manually defined features from the article to be augmented. Instead, it is based on principal component analysis, a well-founded mathematical generalization technique, and predicts new links purely based on the statistical structure of the graph formed by the existing links. Our method does not rely on the textual content of articles; we are exploiting only hyperlinks. A user evaluation of our technique shows that it improves the quality of top link suggestions over the state of the art and that the best predicted links are significantly more valuable than the 'average' link already present in Wikipedia. Beyond link prediction, our algorithm can potentially be used to point out topics an article misses to cover and to cluster articles semantically.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*linguistic processing*; I.5.4 [**Pattern Recognition**]: Applications—*text processing*

General Terms

Algorithms, Experimentation

Keywords

Data Mining, Link Mining, Graph Mining, Wikipedia, Principal Component Analysis

CIKM'09, November 2–6, 2009, Hong Kong, China.
Copyright 2009 ACM 978-1-60558-512-3/09/11 ...$10.00.

1. INTRODUCTION

In 1407, in times of the Ming dynasty, two thousand Chinese scholars assembled the *Yongle Encyclopedia,* a corpus that was to hold the record as the largest coherent repository of human knowledge for 600 years. Things changed when the English version of Wikipedia crossed the two million article mark in 2007. Not only is it now the record-holder in terms of size, it also constitutes a structural paradigm shift. Whereas traditional encyclopedias are sequential, i.e. ordered along alphabetical, topical, or historical lines, Wikipedia has a hypertextual graph structure, in which every article is connected to hundreds or even thousands of other articles by means of hyperlinks placed directly on the words they strive to explain (the so-called *anchors*).

The daunting number of two thousand authors working on the *Yongle Encyclopedia* is also surpassed by Wikipedia, as every Internet user can potentially contribute. To maintain a consistent degree of quality, authors are encouraged to adhere to a *Manual of Style* [19, 18], which stipulates, among many other things, the following:

> "Provide links that aid navigation and understanding, but avoid cluttering the page with obvious, redundant and useless links. An article is said to be *underlinked* if subjects are not linked that are helpful to the understanding of the article or its context. However, *overlinking* is also something to be avoided, as it can make it harder for the reader to identify and follow those links which are likely to be of value." [18]

However, since humans are not flawless and the experience level varies widely among contributors, articles deviate frequently from these rules, which affects the textual content of articles as well as the hyperlinks they comprise. Consequently, human authors often forget to add links that should be there according to the editing guidelines. It would be desirable to detect such missing links automatically because it could enhance the browsing experience significantly. Artificial intelligence and data mining programs that exploit Wikipedia's link structure would equally profit from a data set that has been improved this way.

In this paper we present an algorithm that has the capability of finding missing links in Wikipedia. As an example, consider the article about Karl Marx. It misses essential connections to other relevant articles, for instance it contains no links to Soviet Union or Proletarian revolution. Our method is capable of predicting these links, as

Suggested link target	Anchors	Gain
Socialism	socialist, socialism	1032.9
Soviet Union	Soviet Union	939.7
Democracy	democratic	892.4
Social democracy	Social-Democratic	826.2
Jew	Jewish, Jews	774.9
State	state, states	734.4
Slavery	slavery	726.3
Politics	political, politics	702.3
Proletarian revolution	proletarian revolution	667.8
Property	property, private property	663.4

Table 1: Top 10 suggestions for missing links to be added to the article about Karl Marx. Anchors are phrases on which the link can be placed. 'Gain' is the score for the suggestion (cf. Section 4).

well as others. The top 10 suggestions are listed in Table 1. (The link to Socialism has actually been added to the online Karl Marx article since March 2009, the date of our local working copy of Wikipedia.)

We use a well-known technique called principal component analysis (PCA) in order to enrich existing articles with new links. Our approach can be viewed as using generalization from existing data in order to align articles to a more uniform linking policy. The intuition underlying our work can be described as *cumulative analogy.* Consider for instance Chuuk, Kosrae, Pohnpei, and Yap, the four states forming the Federated States of Micronesia. If most articles that link to Chuuk, Kosrae and Pohnpei also link to Yap, then another article that already links to Chuuk, Kosrae and Pohnpei but not to Yap should probably be modified by adding that missing link—provided the word 'Yap' occurs in the article.

The remainder of this paper is structured as follows. In Section 2 we summarize previous related work in terms of problem domain and methodology. Section 3 introduces Wikipedia's adjacency matrix, the data structure serving as input to our algorithm. Then, in Section 4, we describe dimensionality reduction, and PCA in particular, and give an intuitive explanation of how and why it works in our setting. Section 5 provides the algorithm and describes how we make it computationally tractable given the sheer size of the input matrix. We demonstrate the quality of our approach in Section 6, by showing that it outperforms the previous state of the art in a human user evaluation. Section 7 discusses the differences between our algorithm and previous work, and points out synergetic effects that might result from combining them; we also discuss implications of our work that go beyond the specific problem of link suggestion. In Section 8 we conclude and discuss future research directions.

2. RELATED WORK

There have been several attempts to tackle the problem of suggesting links for Wikipedia.

The closest to our approach is the one developed by Adafre and de Rijke [1]. Like ours, it can enrich articles that already contain some outgoing links and is based on the structure of the Wikipedia link graph. The method consists of two steps. First, it identifies a set of articles which are similar to the input article. Then, the outgoing links that are present in the similar articles but not in the input article are suggested to be added to the input article. A link is only suggested if its anchor text in the similar article is also found in the input article.

In step one, similarity is defined in terms of incoming links. Intuitively, given two articles, if it is often the case that the same page refers to both articles, then the two articles will be considered similar. The actual implementation is more complicated, consisting of several steps harnessing the indexing feature of the custom search engine Lucene [3].

Another, more recent method was proposed by Mihalcea and Csomai [7]. It differs from [1] and the work presented here in that its input is a piece of plain text (the raw content of a Wikipedia article or any other document). It operates in two stages: detection and disambiguation. First, the algorithm decides which phrases should be used as link anchors, then it finds the most appropriate target articles for the link candidates.

To detect link candidates, the best method they tried computes the *link probability* of candidate phrases and selects the top 6% of them. The link probability of an n-gram T is defined as the number of Wikipedia articles containing T as a link anchor divided by the number of articles containing T. It is the prior probability of T being used as a link anchor given that it appears in an article. For instance, the n-gram 'big truck' has a link probability of 0%, whereas 'Internet' has link probability 20%, i.e. every fifth article that mentions the Internet contains a link to its article. In this approach, 'Internet' is likely to be linked again, while 'big truck' is considered to not be a useful link anchorage.

Once the anchors have been chosen, disambiguation is key, since many phrases have several potential meanings. For instance, the phrase 'Monk' will refer most of the time to a male nun and should point to the article Monk, whereas in a jazz-related article, it should probably link to Thelonious Monk. To decide the best sense of a phrase, Mihalcea and Csomai extract local features from surrounding text and train a machine learning classifier from Wikipedia articles, which can serve as labeled examples since the links they contain are already disambiguated. The features are a set of words occurring frequently in the document, as well as the three words to the left of the candidate, the three words to its right, and their parts of speech.

A third method, proposed by Milne and Witten [10], consists of the same steps, but in swapped order. They first find the best sense of each phrase and only then decide which phrase to use as a link anchor.

To disambiguate a term, they look up the articles to which it points when it occurs as a link anchor in Wikipedia. They call the frequency of each potential target article (or sense) its 'commonness'. Then they find all the unambiguous terms in the document; these are the terms that link to the same target article regardless where they occur as anchors in Wikipedia. Then they compute the average semantic 'relatedness' between the candidate term and the unambiguous terms. While any relatedness measure could be plugged in, theirs is itself derived from Wikipedia [9]. Finally, they train a machine learning classifier to combine commonness and relatedness and predict the most appropriate sense of each phrase.

After all phrases have been disambiguated, Milne and

Witten decide which of them to use as link anchors, based on several features of the input article. These include, among others, the link probability of the candidate, its semantic relatedness to the context, how often it appears in the document, and in what positions. Again, all features are combined to train a machine learning classifier. This approach outperforms the predecessor due to Mihalcea and Csomai.

Our technique is different from the two outlined last in that it strives to complete documents that already contain some links to Wikipedia. The input document may be a Wikipedia article or, alternatively, a piece of text that has been 'preprocessed' by one of the above methods. In this paper, we concentrate on the former case.

While our paper and the approaches just summarized deal with very similar problems, the methodology we propose is rather different. We draw on work done by the commonsense reasoning community, most notably the paradigm of cumulative analogy explained in the introduction, which is due to Chklovski [5]. There, the goal is to infer new commonsense facts by analogy. A typical example would be: 'I know many things with feathers, wings, and a beak. So a new thing I find that has feathers and wings probably has a beak as well.' Chklovski proposed 'hand-coded' rules for the cumulative analogy heuristic. Speer *et al.* [13] implemented this type of reasoning automatically by doing principal component analysis. Our technique is inspired by theirs, but we transfer it to the novel domain of link mining.

3. WIKIPEDIA'S ADJACENCY MATRIX

The hyperlink structure of Wikipedia is captured completely in its adjacency matrix. Let N be the number of articles. Then the adjacency matrix has N rows and N columns. The entry at position (i, j) is 1 if article i has a link to article j and 0 otherwise.

In this work we modify the adjacency matrix in two ways. First, we weight columns according to how many articles link to the respective article. This is useful because links pointing to an article that is rarely linked are more informative than links to articles that are linked from nearly everywhere else. For instance, around 320,000 articles link to United States of America, while only 500 link to Federated States of Micronesia. The fact that an article links to Federated States of Micronesia is much more characteristic than that it links to United States of America. In particular, we use the weighting scheme of [9], as follows. Let $\mathbf{W}$ be the weighted adjacency matrix. Its value at position (i, j) is

$$w_{ij} = \begin{cases} -\log(d_j/N) & \text{if article } i \text{ links to article } j, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where d_j is the number of articles containing a link to j. Thus, the term $-\log(d_j/N)$ is the information content of the event 'picking an article that links to j' when we draw a Wikipedia article uniformly at random.

After weighting columns this way, we center $\mathbf{W}$ around the mean by subtracting the respective column mean from each column. This results in the weighted, mean-centered adjacency matrix $\mathbf{A}$, whose columns all have mean 0, a technical requirement of the mathematical methods we are using.

4. DIMENSIONALITY REDUCTION

In this section we will show how the reasoning scheme of cumulative analogy, which we presented in the introduction, is elegantly implemented by principal component analysis (PCA) [11], without any need for coding it explicitly. In order to make the intuition clear, we will first give a brief summary of how PCA works in the context of the Wikipedia adjacency matrix.

In PCA, the rows of the input matrix $\mathbf{A}$ are considered data points, while the columns are taken as features. Therefore, an article (a row in $\mathbf{A}$) is characterized by its links to all the other articles, which makes the outlinks the features of the input matrix. The articles form a cloud of points in an N-dimensional vector space (let us call it *article space*). Since $\mathbf{A}$ is mean-centered, the cloud is centered around the *average article* sitting in the origin.

This cloud is not uniformly distributed but rather sprawling in certain directions and squished in others. This is due to correlations among the points: articles that all link to one specific article often have other particular outlinks in common. For example, articles with links to Adam will often link to Eve as well. PCA finds the directions along which the point cloud is spread out most, i.e. along which articles tend to differ most from the average article. Those directions are called *principal components.* They are vectors in the N-dimensional article space pointing away from the average article in the origin; by convention, they are normalized to a length of 1.

The principal components found are orthogonal. Hence, an appealing geometric way of thinking about PCA is as a rotation of the axes of the coordinate system such that the spread (more formally, the variance) of the data is k-th largest along dimension k; it then computes the co-ordinates of each point in the new basis formed by the principal components. The principal components themselves can be considered 'fantasized' articles (since they are points in the N-dimensional article space).

Mathematically, the principal components are the eigenvectors of the data covariance matrix. Hence, we call them *eigenarticles,* to emphasize that they are eigenvectors and points in article space. The new space resulting from the rotation is called *eigenspace.*

Computing the coordinates of an article $\mathbf{a}_i$ (a row vector of $\mathbf{A}$) in eigenspace amounts to projecting it onto the eigenspace basis vectors, i.e. onto the eigenarticles $\mathbf{e}_k$. Then the vector $\mathbf{p}_i$ of projections is the eigenspace representation of $\mathbf{a}_i$:

$$\mathbf{p}_i = (p_{i1}, ..., p_{iN}) = (\mathbf{a}_i\mathbf{e}_1^T, ..., \mathbf{a}_i\mathbf{e}_N^T) \tag{2}$$

In matrix notation this can be written succinctly as

$$\mathbf{P} = \mathbf{A}\mathbf{E}^T, \tag{3}$$

where projection vector $\mathbf{p}_i$ is the i-th row of $\mathbf{P}$ and eigenarticle $\mathbf{e}_k$ is the k-th row of $\mathbf{E}$.

Since PCA performs a rotation, $\mathbf{E}^T$ is a rotation matrix, i.e. $\mathbf{E}^T\mathbf{E}$ is the identity matrix. Thus, the reverse projection from eigenspace back into article space (the so-called reconstruction of $\mathbf{A}$) is

$$\mathbf{A} = \mathbf{P}\mathbf{E}. \tag{4}$$

Expanding (4), a single entry of $\mathbf{A}$ is computed in the reconstruction as follows:

$$a_{ij} = \sum_{k=1}^{N} p_{ik}e_{kj}. \tag{5}$$

Entry a_{ij} is large when there are many eigenarticles $\mathbf{e}_k$ that (a) are important components of $\mathbf{a}_i$ in eigenspace (resulting in large p_{ik}) and that (b) link themselves to article j (resulting in large e_{kj}). Remember that eigenarticles live in article space and have 'fantasized' outlinks to 'real' articles, e_{kj} being the weight of the k-th eigenarticle's link to article j.

The reconstruction of $\mathbf{A}$ as $\mathbf{PE} = \mathbf{AE}^T\mathbf{E}$ is exact. However, getting an exact reconstruction is not useful from our point of view. What we would like, intuitively, is to find first the articles similar to the article of interest. Then, we would like to propose *new links*, which are not present in the original matrix $\mathbf{A}$, but which are suggested because they appear in similar articles. This is an important difference compared to more traditional applications of PCA, in which one wants to obtain a reconstruction that is as exact as possible. We actually want to obtain a reconstruction that enriches the original data.

In order to achieve this goal, we must first ensure that we 'forget' some information while dwelling in eigenspace, just like in the case of traditional dimensionality reduction. First, we project an article $\mathbf{a}_i$ from article space into eigenspace, obtaining its eigenspace representation $\mathbf{p}_i$ (cf. (2)). Now we 'shrink' $\mathbf{p}_i$ by setting to zero all components p_{ik} with $k > K$ for some fixed K. These were the projections onto eigenarticles along whose direction the variation in the data is small, so it can be considered noise. By shrinking $\mathbf{p}_i$ we eliminate that noise. Now we can reconstruct $\mathbf{a}_i$ approximately by projecting it back into article space (cf. (4)):

$$\mathbf{A}_K = \mathbf{P}_K\mathbf{E}_K, \tag{6}$$

where $\mathbf{P}_K$ consists of the first K columns of $\mathbf{P}$ and $\mathbf{E}_K$ of the first K rows of $\mathbf{E}$. Matrix $\mathbf{A}_K$ still has the same dimensions as $\mathbf{A}$, but its entries have changed values, since (6) amounts to replacing N with K in (5):

$$a_{ij}^K = \sum_{k=1}^{K} p_{ik}e_{kj} \tag{7}$$

After the back-projection we compare a_{ij} to a_{ij}^K. If there was no link between articles i and j originally, but $a_{ij}^K \gg a_{ij}$, then our method predicts that the link should be added.

To see how this algorithm naturally incorporates the cumulative analogy scheme, let us look at the system in action. Consider an article i which should link to article j but does not. Now consider also a set of articles $\mathcal{A}$ which are similar to article i, in terms of the other outgoing links. These articles will reside in a part of article space similar to i, so they will project similarly onto eigenarticles (because a rotation will preserve the neighborhood structure of these articles). If many of the articles in $\mathcal{A}$ contain j as an outlink, the eigenarticles on which they cause a significant projection will also link to j. These eigenarticles will cause the value a_{ij}^K to increase, compared to a_{ij}, so article j will be suggested as a link from i as well. Its absence in the original article is, in this case, directly attributed by our method to noise, caused by projecting onto insignificant eigenarticles.

Note that no heuristic is involved in our method. It simply exploits the statistical properties of the set of already existing links. We emphasize again the particular flavor of the use of PCA here (as also in [13]). Typical PCA applications strive to minimize the reconstruction error while compressing the data through dimensionality reduction (e.g. [14]). In our paradigm, the 'error' is exactly what we want! To underline this, we should speak of *reconstruction gain* or generalization gain rather than reconstruction error.

Algorithm 1 Wikipedia link suggestion

```
Input: Article $i$, represented by its outlinks $\mathbf{a}_i$;
  minimum link probability $\alpha$
Output: Link suggestions for article $i$, in order of decreasing quality
Static: Eigenarticle matrix $\mathbf{E}_K$
  $\mathbf{p}_i \leftarrow \mathbf{a}_i\mathbf{E}_K^T$ (projection into reduced eigenspace)
  $\mathbf{a}_i^K \leftarrow \mathbf{p}_i\mathbf{E}_K$ (projection back into article space)
  $\mathbf{g}_i \leftarrow \mathbf{a}_i^K - \mathbf{a}_i$ (the reconstruction gain vector)
  $\mathcal{S} \leftarrow \emptyset$ (set of link candidates)
  for $n$-grams $T$ of text of article $i$ do
    if $T$ has link probability $> \alpha$ and there is an article $j$
    about topic $T$ and $i$ has no link to $j$ then
      Add $j$ to $\mathcal{S}$
    end if
  end for
  for $j \in \mathcal{S}$, in order of descending $g_{ij}$ do
    Suggest link from $i$ to $j$
  end for
```

5. IMPLEMENTATION

Pseudocode for the method we just described is provided in Algorithm 1. The steps laid out in Section 4 are followed directly. The article to be augmented is projected into the reduced eigenspace, then back into article space. The output is a list of link suggestions, ordered by the reconstruction gain of the links, i.e. by how much more weight they have after the projections than before.

Of course, a link can be suggested only if the appropriate anchor term occurs in the text of the source article. In order to prune away nonsense terms and stopwords from the beginning, and thus speed up the algorithm, we consider as potential anchors only n-grams whose link probability (cf. Section 2) is above a specified threshold α. A value of $\alpha = 6.5\%$ was found to afford optimal performance [10], which is why we use this threshold in our implementation.

We ran our algorithm on two versions of Wikipedia: one consisting of a carefully selected small subset of important articles, the other being a recent data dump of the entire Wikipedia. The complete version is three orders of magnitude larger than the small selection, so we had to apply additional tricks in order to make PCA computationally tractable on the former. We now describe both our implementations.

5.1 Wikipedia Selection for Schools

The '2008/9 Wikipedia DVD Selection is a free, hand-checked, non-commercial selection from Wikipedia, targeted around the UK National Curriculum and useful for much of the English speaking world.' [16] It is edited by SOS Children's Villages UK and contains 5,503 articles (so $N = 5{,}503$) that can serve as a free alternative to costly encyclopedias. As most Wikipedia articles are not present in it, the majority of links had to be removed, too. All links pointing to articles included in the collection were kept; redirects were resolved (e.g. links to MÜNCHEN were changed to MUNICH, since the two are different titles of the same article) [4].

Using Matlab's built-in functions, computing the eigenarticles and implementing Algorithm 1 was straighforward.

For this data set we chose an eigenspace dimensionality of $K = 256$. We will describe the results in Section 6.2.

5.2 Full Wikipedia

While the Wikipedia Selection for schools serves well as a proof of concept and for evaluating the potential of the technique, the full version of Wikipedia is certainly more interesting, for several reasons.

First, Wikipedia's live online version is consulted by many Internet users on a daily basis. So, if our method can improve full Wikipedia, it will have much more traction than if it worked only on a small subset of articles.

Second, live Wikipedia is evolving constantly, articles being added or modified constantly. Thus, if our method is applicable to full Wikipedia, then it can be used by authors every day to find links they have probably forgotten to include in the articles they are writing.

Third, Wikipedia contains over two million articles (three orders of magnitude more than the school selection). In order to cope with such a challenging amount of information, our algorithm really has to scale well.

Fourth, previous methods used full Wikipedia as a data set, and we want to compare the performance of our technique directly to them.

We downloaded the Wikipedia snapshot of March 6, 2009 [17], and indexed it in a database to facilitate quick look-up of basic information such as the set of links contained in an article or pointing to it. The database contains 20 GB of information. To create, fill and access it, we used a Java toolkit called WikipediaMiner [8], written by David Milne.

The data dump contains over six million pages, 2,697,268 of which are actual articles (the rest are, among others, category, redirect, or disambiguation pages). Now $N = 2{,}697{,}268$ and consequently the $N \times N$ adjacency matrix $\mathbf{A}$ would occupy 29 terabytes of memory (assuming 32-bit floating point precision); a sparse representation is useless as well, because mean-centering turns most zeros of the sparse original adjacency matrix into negative numbers.

So, in order to make PCA and thus our method tractable on full Wikipedia, we have to apply some further tricks.

First, we reduce the size of $\mathbf{W}$, the weighted, non-mean-centered adjacency matrix. In terms of columns, we keep only those associated with articles that have at least 15 incoming and 15 outgoing links. This way we eliminate articles about the most obscure topics—seemingly a majority of Wikipedia—, reducing the width of the data matrix to $w = 468{,}510$ (17% of the original width). The same method of constraining the set of articles is used by [6]. Remember that columns are the features of the data matrix, so discarding 83% of the columns could be described as feature selection.

To compress the height of $\mathbf{W}$, we keep a row only if the article it represents is about a topic for which the schools selection contains an article as well. This reduces the height of the data matrix to $h = 5{,}503$ (0.2% of the original height). Recall that rows are the data points of the data matrix, so discarding 99.8% of the rows amounts to shrinking the set of training samples for our algorithm aggressively, to only the most important articles (as determined by this other source of information). We will show in Section 6.1 that restricting the set of training articles that drastically does not impede our ability to suggest links for new articles that have not been encountered during training.

After decreasing the size of $\mathbf{W}$, we mean-center it and obtain the $h \times w$ matrix $\hat{\mathbf{A}}$. This matrix has a lot more columns than rows, which makes it amenable to a trick used in a seminal image processing paper on 'eigenfaces' [14]. As mentioned in Section 4, the eigenarticles are the eigenvectors of the data covariance matrix, which can be written as $\hat{\mathbf{A}}^T\hat{\mathbf{A}}$. By definition, this means

$$\hat{\mathbf{A}}^T\hat{\mathbf{A}}\mathbf{e}_k = \lambda_k \mathbf{e}_k, \tag{8}$$

for an eigenarticle $\mathbf{e}_k$ with associated eigenvalue λ_k.

Now consider the eigenvectors of another matrix, $\hat{\mathbf{A}}\hat{\mathbf{A}}^T$. Eigenvector $\mathbf{v}_k$ fulfills

$$\hat{\mathbf{A}}\hat{\mathbf{A}}^T\mathbf{v}_k = \mu_k \mathbf{v}_k, \tag{9}$$

for eigenvalue μ_k. Left-multiplying by $\hat{\mathbf{A}}^T$ yields

$$\hat{\mathbf{A}}^T\hat{\mathbf{A}}(\hat{\mathbf{A}}^T\mathbf{v}_k) = \mu_k(\hat{\mathbf{A}}^T\mathbf{v}_k). \tag{10}$$

So each $\hat{\mathbf{A}}^T\mathbf{v}_k$ is an eigenvector of $\hat{\mathbf{A}}^T\hat{\mathbf{A}}$. More precisely

$$\mathbf{e}_k = \hat{\mathbf{A}}^T\mathbf{v}_k, \quad \lambda_k = \mu_k. \tag{11}$$

The crucial observation is that $\hat{\mathbf{A}}\hat{\mathbf{A}}^T$ is $h \times h$, i.e. 5,503 × 5,503 in our case, which means it fits into memory, making it possible to compute the eigenvectors $\mathbf{v}_k$ efficiently. Subsequently, we can find eigenarticle $\mathbf{e}_k$ simply as $\hat{\mathbf{A}}^T\mathbf{v}_k$.

Once the eigenarticles have been computed (we used eigenspace dimensionality $K = 1{,}000$ for the full Wikipedia data set), Algorithm 1 can be deployed just as for the small schools selection.

6. EVALUATION

6.1 Full Wikipedia

We evaluated the performance of our link suggestion algorithm by querying human raters on Amazon Mechanical Turk [2]. Mechanical Turk is an online platform on which 'requesters' can post questionnaires (among many other types of tasks), which are subsequently completed for a typically small amount of money by 'workers', regular Internet users who have registered with the system. It has recently been shown that non-expert labels obtained through Mechanical Turk agree very well with gold-standard expert annotations for natural language tasks [12], which justifies using it for our purpose.

In each rating task we presented the human contributor with the text of a randomly selected Wikipedia article about a topic T. The article text still contained the original outgoing links. The task description read as follows:

> "You are presented with the text of a Wikipedia article about T.
>
> Below the article text, you are given the titles of four other Wikipedia articles. The article about T could potentially contain a link to each of these four articles.
>
> Your task is to identify the one link (from the list of four) which you consider most useful. A useful link should lead to an article that is relevant for the article about T, and which readers of the article about T would likely want to investigate further.

> In case you are not familiar with T, please make sure you get an idea of who or what T is by looking through the article text."

In order to be able to compare our algorithm to Milne and Witten's, the definition of a useful link is directly copied from their instructions to human raters [10], which in turn capture Wikipedia's linking policy [19].

The four outgoing links between which raters had to choose were the following.

1. The top link suggestion S made by our method, using the $K = 1{,}000$ most significant eigenarticles. Note that the article always contained an appropriate anchor for suggestion S, and of course T itself was never chosen as a suggestion.
2. The top link suggestion S_{mw} made by Milne and Witten [10], i.e. the one to which their system attributes the highest confidence value. Their code is included in the WikipediaMiner toolkit [8] and could thus be used as a black box.
3. A pre-existing link S_{pre} already present in article T, selected uniformly at random, but different from S and S_{mw}.
4. A link S_{rnd} to an article that is not linked from T but that could potentially be linked because its title is one of the n-grams of T's plain text (as n-grams we chose all sequences of between one and four words). Again, this is chosen randomly and different from S and S_{mw} (and from S_{pre} by definition). This serves as a random baseline.

The order of the four choices was randomized, to prevent any bias.

We evaluated the performance on a set of 181 articles randomly picked from the set of articles not used in computing the eigenarticles, to avoid overfitting and test whether our algorithm generalizes well to unseen data; call this set the *test set*. We constrained our random selection to articles with at least 100 incoming and at least 100 outgoing links. The reasoning is similar to that behind our choice of the columns of $\hat{\mathbf{A}}$ (cf. Section 5.2): we wanted to ensure that the articles were not about very obscure topics, so human raters would not have to read the article text in depth to be able to make an informed decision.

To facilitate the performance analysis, we considered only articles on which our method and that of Milne and Witten did not agree. (Out of the 200 articles we initially tried, the methods agreed on 8%.)

Each task was completed by six different raters, so we gathered $6 \times 181 = 1{,}086$ votes. As a safeguard against participants who might potentially have clicked randomly rather than made an informed decision, we implemented a voting scheme that counts a vote only if it agrees with at least two others on the same task, which resulted in a set of 660 effective votes. This heuristic is justified *a posteriori* by the low performance of the random baseline, which is according to our expectations.

The results are summarized in Figure 1 (all gathered data can be found online [15]). Our method won most votes (36%), followed by Milne and Witten (27%), the random pre-existing links (25%), and finally the baseline of random n-grams (11%).

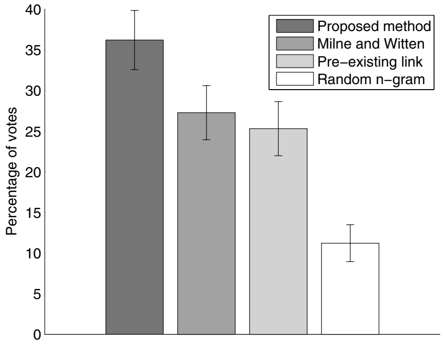

Figure 1: Results of the human user evaluation, in terms of percentages of votes won by the different link types (explained in Section 6.1). The error bars show the 95% confidence intervals.

Thus, our method outperforms the previous state of the art. Our top suggestion is considered best 9% more often than theirs. A difference of at least 4% is statistically significant at the $p < 0.05$ level (estimated by bootstrap resampling).

Also, the fact that our suggestions won significantly more votes than the randomly picked pre-existing links (11% difference; at least 6% is significant at the $p < 0.05$ level) implies that the top links our method finds are better than the average human-added link: we do not just find minor links that happen to have some relevance for the article being augmented; instead, we find important links that the human authors forgot to include.

This quality of suggestions is reached on a set of test articles that did not partake in the eigenarticle calculation, which implies that our algorithm generalizes well to articles it was not trained from. This is crucial because it justifies selecting only a small subset of all Wikipedia articles as rows of $\hat{\mathbf{A}}$ (cf. Section 5.2), a restriction without which PCA on the enormous adjacency matrix would be computationally infeasible.

6.2 Wikipedia Selection for Schools

To illustrate the effect of our technique on more than a few hand-picked examples, we augmented a complete local copy of the 2008/9 Wikipedia Selection for schools by adding the 17,000 highest ranking links suggested by Algorithm 1 (an average of approximately three new links per article). The result can be browsed online [15].

Since we have already shown that our algorithm performs very well on the full version of Wikipedia, we do not evaluate the quality of link suggestions formally on the small selection as well. Instead, we focus on a more qualitative analysis. In particular, a desirable property of the algorithm, which was not measured in the previous set of results, would be a decay in quality with the ranking of the suggestion; i.e. we would expect links with small reconstruction gain to be less useful. Figure 2 suggests that this is the case. To make the argument clear, we point out that the first **for** loop of Algorithm 1 considers only link candidates that could potentially be accepted because an appropriate anchor appears in the text of the source article. This is exclusively for reasons of

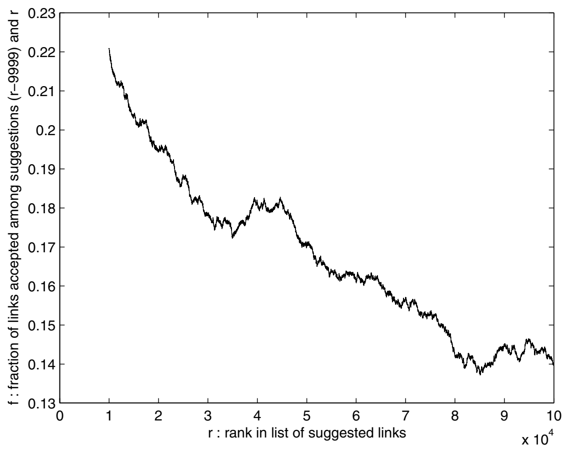

Figure 2: Running average of the number of suggestions that are acceptable because an appropriate anchor for the target appears in the source article. Note that the maximum is deceptively low because the running average is taken over 10,000 consecutive ranks.

efficiency. We might just as well loop over *all* potential target articles, regardless of whether there is an apt anchor or not. This way we can first collect all link predictions and calculate later for how many of them a fitting anchor exists.

Figure 2 plots the running average of this quantity as a function of rank in the list of link suggestions. The r-axis shows the rank; the f-axis shows the fraction of suggestions for which an anchor exists in the source article, among the 10,000 suggestions up to rank r. The fact that this fraction decays as we descend in the list of suggestions means that fewer and fewer of the predicted links have an anchor in the source article, which in turn implies that the quality of suggestions decays as well: the less likely a target article is to have an anchor in the source article, the less likely it is to be a valuable suggestion. Our algorithm does not just roughly separate good from bad suggestions, it also ranks them continuously in a sensible way.

Note that the probability of a random article name appearing in the text of another random article is only 1.5% (estimated from 10,000 randomly selected article pairs), significantly lower than the 14% that Figure 2 shows for suggestions 90,001 to 100,000. This means that not only our *top* suggestions are much better than random ones (as shown in Section 6.1) but that this is true even far down in our ranking.

7. DISCUSSION

7.1 Comparison to Previous Methods

To highlight the contributions of this research, we will now contrast it with the existing methods referenced in Section 2.

As mentioned, the technique coming closest to ours is that of Adafre and de Rijke [1], since it is based on the links rather than the text that articles contain. However, there are several important differences.

Adafre and de Rijke gauge the similarity of two articles in terms of how many incoming links they share. To augment an input article with new links, they copy links from any *single* article that is sufficiently similar to the input article according to this measure. Our method represents articles in terms of their outgoing links. This makes it possible to apply the cumulative analogy paradigm: 'If there are *many* articles sharing a lot of features (outlinks) among each other and with the input article, and if these articles also share a certain single feature (outlink), then the input article should have that feature (outlink), too.' The fact that many similar articles, rather than just a single one, are required makes the method more robust to noise.

In addition to this robustness concerning where to copy from, our technique is also more careful regarding what to copy. If an article is similar enough to the input article, Adafre and de Rijke copy any of its outgoing links, as long as the appropriate anchor text occurs in the input article. On the contrary, our method works with numerical values and can thus weight outlinks with importance values (reconstruction gain).

Also, the approach we propose naturally incorporates the two steps (picking the similar articles and ranking the candidate links before suggesting them for the input article) into one simple mathematical operation: PCA. Adafre and de Rijke's first step alone seems considerably more complicated, involving a scheme of several rounds of querying the search engine that indexes the incoming links of each article.

Before we compare our method to [7] and [10], we will first summarize their principal properties (for more details, see Section 2) in a concise list:

1. Both methods consist of **two separate phases**, link detection and link disambiguation.

2. They rely heavily on **several hand-picked features**, used to train machine learning classifiers.

3. These features strive to capture the **textual content** of the article to be augmented.

We demonstrated that we can outperform the state of the art [10] with an algorithm that elegantly integrates detection and disambiguation in one **single phase**. To illustrate this, it is worthwhile to point out a subtlety we have glossed over in the pseudocode of Algorithm 1. We wrote 'topic T' in the first loop, while in fact T is an n-gram, i.e. a sequence of words, which could be ambiguous. However, mapping the n-gram to the most appropriate target article is easy: given a source article i, the PCA will already have computed a score (the reconstruction gain) for *every* other Wikipedia article, so to retrieve the most appropriate sense of the n-gram T in article i, we simply look at all possible senses (all articles the anchor T ever links to in all of Wikipedia) and define 'topic T' as the one with highest reconstruction gain for source article i.

Even if the features used in the two approaches make sense intuitively and turn out to work well, they still had to be defined 'manually' by experts. On the contrary, our method is **featureless**. It merely completes the hyperlink structure of a document collection by means of a mathematically sound and proven generalization technique. There is no need to 'force' the algorithm to follow Wikipedia's linking policy [19] by hand-crafting features that more or less encode those rules. Our technique starts from whatever linking policy is in place—most articles abide by it very closely to begin with!—and enforces it where it is infringed, by eliminiating the noise such a deviation represents.

The algorithm we propose works on the **hypertextual content** of an article (the set of outgoing links), not on its raw text. No advanced scanning or even parsing is necessary, as in the two text-based methods (e.g. Milne and Witten [10] need to know at what position in the article a phrase occurs; Mihalcea and Csomai [7] even require part-of-speech tagging). We only ever inspect the article content in one trivial way, to see which n-grams it contains (and once, offline, to calculate link probabilities; but as explained in Section 5, this is not even integral to our approach but just a means of speeding up the algorithm). Our technique is based entirely on the link structure of the document collection. This rich source of information is left completely untapped by Mihalcea and Csomai. Milne and Witten do use link structure, but more indirectly, to compute their semantic relatedness measure. However, since it is used as a black box, this component could be replaced with any such measure and is not an integral ingredient of their approach.

It should be mentioned that the memory requirements of our algorithm can be rather high, depending on how one chooses the eigenspace dimensionality K, since the eigenarticles have to be stored in RAM. Memory usage grows linearly in K.

7.2 Synergies

Content-based methods have the advantage of being able to add links to raw text rather than documents that already come with a set of Wikipedia links. This is why it is important to point out that in the big picture our algorithm is not so much a rival of [7] and [10] as rather a tool to exploit dimensions of Wikipedia unaccounted for by those predecessors. Consequently, we conjecture that a combination of textual and hypertextual methods might have a synergetic effect: while in this paper we restricted ourselves to showing that our technique works well for suggesting links within Wikipedia, the method is applicable, without any changes, to any input document containing a basic set of links to Wikipedia. It could thus employ a text-based link suggester such as [7] or [10] as a preprocessor and fill in links those methods have missed. We conducted preliminary experiments with this approach but do not report results, for the lack of a formal evaluation.

One could even go further and couple a textual technique with our hypertextual one, in order to link a complete plain-text document collection (such as a large news story archive) to Wikipedia in three steps: first, add a basic set of links to each document by means of a text-based technique; second, compute the eigenarticles for this document collection; third, run our method on all articles to complete the link structure. Step two only serves the purpose of fine-tuning the method to the characteristics of the document collection at hand. Alternatively, the eigenarticles computed from Wikipedia can be used.

A synergetic effect may also be expected when our method is deployed in a feedback loop. As Wikipedia authors accept (or reject) an increasing number of link suggestions, Wikipedia will comply ever closer to its own linking policy, which in turn means more accurate training data for the next generation of suggestions. A similar argument could be made for the text-based methods, yet it is more immediate for our approach, since it takes its own output—link structure—directly as input.

	Suggested link target	**Gain**
	Random variable	3.232
★	Variance	2.819
	Probability distribution	2.469
	Median	1.800
	Real number	1.454
	Poisson distribution	1.450
	Exponential distribution	1.447
	Binomial distribution	1.385
	Chi-square distribution	1.353
	Psychology	1.145
	Physics	1.079
★	Engineering	1.031
★	Economics	1.018
	Computer science	0.991
	Arithmetic mean	0.926

Table 2: Top 15 suggestions of Algorithm 1 for links to be added to the article about Statistics in the 2008/9 Wikipedia Selection for schools. 'Gain' refers to reconstruction gain. Links marked with a star could actually be added because the appropriate anchor text occurred in the source article.

7.3 Detection of Missing Topics

While link suggestion is useful in its own right, the reach of our technique goes beyond. Recall that, unlike the existing approaches, our algorithm computes scores not only for phrases appearing in the input article but for *every* Wikipedia article. Let us take Table 2 as an example. It shows the top 15 suggestions of Algorithm 1 for the Statistics article of the Wikipedia Selection for schools. Note that many links (those not marked with a star) could not be suggested for the sole reason that there was no appropriate anchor text in the source article. It is interesting to see that, more often than not, it would be desirable if the article about Statistics did in fact cover the target topic. For instance, it is well possible that the author simply forgot to properly introduce the concepts Random variable and Probability distribution or to mention that Statistics is of foremost importance to modern Physics.

Consequently, our method can be deployed not only to suggest missing links but also to suggest missing topics. This feature, too, distinguishes our method significantly from previous link suggestion methods [7, 10]. They constrain their suggestions to topics that are present in the source article in the first place, and are thus unable to predict which topics *should* be present. They can only decide whether a term that already appears in the article text should be used as a link anchor. Previous methods are topic detectors, ours is at heart a topic suggester.[1]

7.4 Concept Clustering

The central computation of our algorithm is the projection of an article onto the eigenarticles. To understand the effect of this operation graphically, let us take a quick peek into eigenspace. Figure 3 plots 200 articles selected ran-

[1] Although Adafre and de Rijke do not mention it, we conjecture that their technique, too, is in principle able to suggest topics.

domly from the full Wikipedia version, neglecting all higher dimensions and showing only the projections onto the two most important eigenarticles. In the notation of Section 4, the axes of the plot are $\mathbf{e}_1$ and $\mathbf{e}_2$, and article i has coordinates (p_{i1}, p_{i2}). The dashed line shows that the plane spanned by $\mathbf{e}_1$ and $\mathbf{e}_2$ is 'semantically separable': articles below the line are nearly exclusively about science-related topics, whereas those above the line live in the realm of the arts and humanities (history, culture, etc.).

This is a consequence of the fact that PCA finds the directions of largest variance in the data. Since a data point is defined by the outgoing links of an article and since articles about science topics typically have a very different set of outlinks from articles about the arts and humanities, these two classes are far apart in the subspace spanned by the first principal components of the data.

These observations suggest that our method may also be used to cluster concepts into semantic classes. Milne and Witten [9] use Wikipedia's link structure to compute the semantic relatedness of pairs of concepts, but they do not include the crucial step of dimensionality reduction, which makes clustering possible by grouping related concepts in a low-dimensional subspace of the original data.

Both the detection of missing topics and semantic concept clustering are currently investigated by our group.

8. CONCLUSION

In this paper we present a novel approach to find missing links in document collections such as Wikipedia. We use exclusively the structure of Wikipedia's hyperlink graph, in a featureless approach based on principal component analysis, a mathematically sound generalization technique. It enforces the linking policy that is implicit in the entirety of Wikipedia's hyperlink structure by putting additional links into those articles that contravene the linking guidelines. The method is conceptually clean, yet its simplicity does not keep it from outperforming the state of the art.

Our method draws on work done by the commonsense reasoning community, and we strive to give an intuitive explanation of how and why it implements the paradigm of cumulative analogy by performing dimensionality reduction. We point out implications of the approach beyond link completion: It can detect topics a given Wikipedia article fails to cover, and cluster articles along semantic lines. We hope this work will inspire the application of similar techniques to other problems in graph and especially Wikipedia mining.

9. ACKNOWLEDGMENTS

The Natural Sciences and Engineering Research Council of Canada (NSERC) supported this research financially. We would also like to thank David Milne for making the WikipediaMiner code publicly available.

10. REFERENCES

[1] S. F. Adafre and M. de Rijke. Discovering missing links in Wikipedia. In *Proc. 3rd Int'l Workshop on Link Discovery (LinkKDD-05)*, 2005.

[2] Amazon. Amazon Mechanical Turk. Website, 2009. `http://www.mturk.com`.

[3] Apache. Lucene. Website, 2009. `http://lucene.apache.org`.

[4] A. Cates. SOS Children's Villages UK. Personal communication, 2009.

[5] T. Chklovski. Learner: A system for acquiring commonsense knowledge by analogy. In *Proc. 2nd Int'l Conf. on Knowledge Capture (K-CAP-03)*, 2003.

[6] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proc. 19th Int'l Joint Conf. on Artificial Intelligence (IJCAI-07)*, 2007.

[7] R. Mihalcea and A. Csomai. Wikify! Linking documents to encyclopedic knowledge. In *Proc. 16th ACM Conf. on Information and Knowledge Management (CIKM-07)*, 2007.

[8] D. Milne. WikipediaMiner toolkit. Website, 2009. `http://wikipedia-miner.sourceforge.net` (accessed June 6, 2009).

[9] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proc. 1st AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI-08)*, 2008.

[10] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *Proc. 17th ACM Conf. on Information and Knowledge Management (CIKM-08)*, 2008.

[11] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[12] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—But is it good? Evaluating non-expert annotations for natural language tasks. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP-08)*, 2008.

[13] R. Speer, C. Havasi, and H. Lieberman. AnalogySpace: Reducing the dimensionality of common sense knowledge. In *Proc. 23rd Nat. Conf. on Artificial Intelligence (AAAI-08)*, 2008.

[14] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[15] R. West. Project website, 2009. `http://www.cs.mcgill.ca/~rwest/link-suggestion`.

[16] Wikipedia. 2008/9 Wikipedia Selection for schools. Website, 2008. `http://schools-wikipedia.org` (accessed June 3, 2009).

[17] Wikipedia. Data dump of March 6, 2009. Website, 2009. `http://download.wikimedia.org/enwiki/20090306` (accessed June 3, 2009).

[18] Wikipedia. Wikipedia:Linking. Website, 2009. `http://en.wikipedia.org/w/index.php?title=Wikipedia:Linking&oldid=304705797` (accessed July 30, 2009).

[19] Wikipedia. Wikipedia:Manual of Style. Website, 2009. `http://en.wikipedia.org/w/index.php?title=Wikipedia:Manual_of_Style&oldid=294792091#Wikilinks` (accessed June 6, 2009).

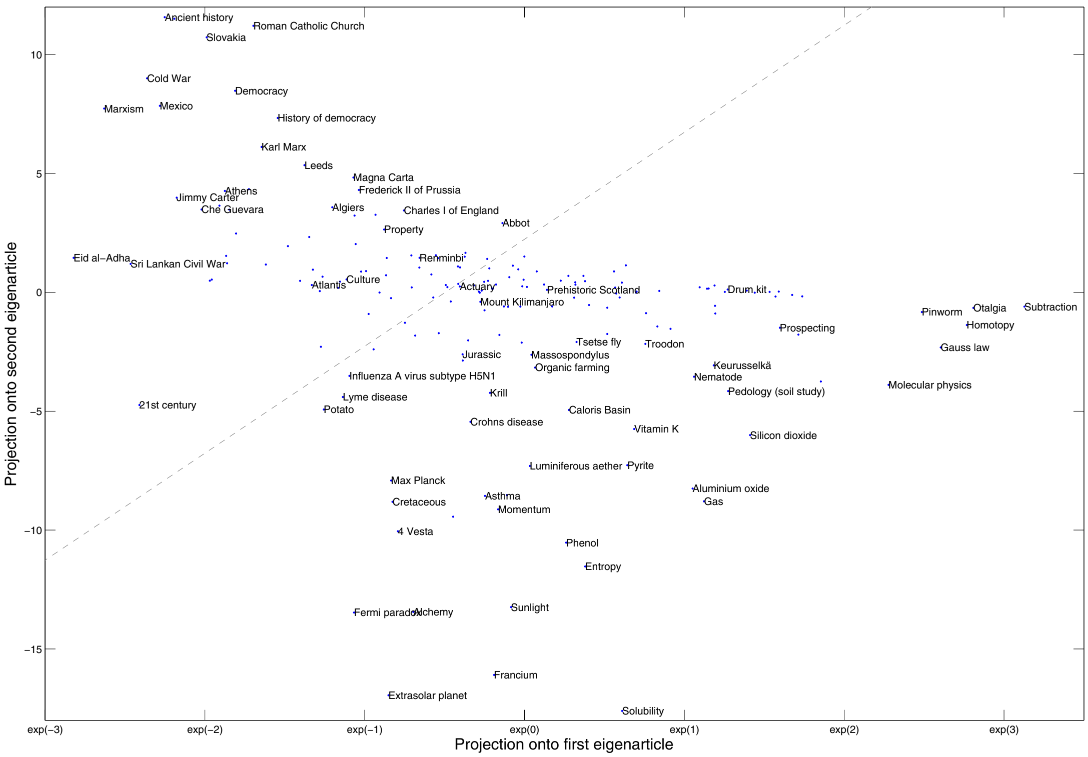

Figure 3: Projection of 200 randomly selected articles onto the two principal eigenarticles. To increase legibility, only a subset of points is labeled, and the x-axis is logarithmic (no log transformation could be performed on the y-axis, since the logarithm is not defined for the negative values). The dashed line roughly separates articles about the sciences from those about the arts and humanities.